

# OpenVMS Logische Namen und Verzeichnisstrukturen

Karl Bruns  
Trainer/Consultant  
OpenVMS and Networking  
OSI, DECnet, X.25 and TCP/IP  
Lessingstr. 1  
D-86438 Kissing  
Phone +49/8233/2938  
Mobile +49/1717168148  
[www.kbruns-training.de](http://www.kbruns-training.de)  
[karl.bruns@web.de](mailto:karl.bruns@web.de)

**DECUS IT - Symposium**

**4. April - 8. April 2005**

**Düsseldorf**

Karl Bruns

1

# OpenVMS Logische Namen und Verzeichnisstrukturen

- Logische Namen werden als Variable für physikalische Gerätebezeichnungen, Verzeichnis- und Dateinamen verwendet
- Ziel: unabhängig von festen Zuweisungen
- Bsp.: `SYSSYSDEVICE` zeigt immer auf die Systemplatte, `SY$LOGIN` definiert das Standardverzeichnis
- Anzeige der logischen Namen:

```
$ show logical
```

Karl Bruns

2

## OpenVMS Logische Namen und Verzeichnisstrukturen

- Logische Namen werden in Tabellen verwaltet :
  - Prozesstabellediese Namen gelten nur den den laufenden Prozess, sollten daher in der Login-Prozedur definiert werden von OpenVMS werden definiert:

SY\$OUTPUT  
SY\$INPUT  
SY\$COMMAND  
SY\$ERROR  
TT

SY\$DISK

Karl Bruns

3

## OpenVMS Logische Namen und Verzeichnisstrukturen

- Jobtabelle
  - ein Job ist die Menge eines Hauptprozesses plus sämtlicher Subprozesse,
  - über diese Tabelle kann Prozesskommunikation erfolgendefiniert ist :

SY\$LOGIN

- Gruppentabelle
  - hier werden Namen für die verschiedenen Gruppen aufgebaut,
  - diese Tabelle ist permanent gültig,
  - Privileg: GRPNAM

Karl Bruns

4

## OpenVMS Logische Namen und Verzeichnisstrukturen

- **Systemtabelle**  
diese Tabelle gilt systemweit für alle Benutzer und wird beim Startup aufgebaut  
Privileg: SYSNAM  
Bsp.:  
SY\$\$MANAGER  
SY\$\$SYSTEM  
SY\$\$NODE  
SY\$\$ANNOUNCE  
MOM\$LOAD  
TCPIP\$HOST
- **Clustertabelle**

Karl Bruns

5

## OpenVMS Logische Namen und Verzeichnisstrukturen

- **Schutzmechanismen für logische Namen**
  - usermode            diese Definition gilt nur für das folgende Image
  - supervisor mode    dies ist der Default
  - execmode            besonderer Schutz
  - kernelmode        Schutz für die Tabellen selbst
- **Anzeige:**    \$ show logical/full

Karl Bruns

6

## OpenVMS Logische Namen und Verzeichnisstrukturen

- Definition eines logischen Namens in der Prozesstabelle:

```
$ DEFINE KREIS DKA0:[HUGO.FORTRAN.PROG]KREIS.EXE
```

- Verwendung:

```
$ RUN KREIS
```

- Löschen:

```
$ DEASSIGN KREIS
```

Karl Bruns

7

## OpenVMS Logische Namen und Verzeichnisstrukturen

- **Logischer Aufbau einer Platte:**

eine Platte ist hierarchisch strukturiert, die höchste Ebene bildet das Master File Directory (MFD) mit Dateien, die die Platte zur Verwaltung benötigt,

z.B.: INDEXF.SYS hier stehen sämtliche Dateiheder

darunter befindet sich die Benutzerebene User File Directory (UFD)

und darunter können die Benutzer eigene Verzeichnisse anlegen

Sub File Directory (SFD)

Karl Bruns

8

## OpenVMS Logische Namen und Verzeichnisstrukturen

- Aufbau einer OpenVMS Systemplatte

nach dem Initialisieren der Platte gibt es nur das MFD mit den Verwaltungsdateien  
nach dem Installieren von OpenVMS ist die Systemplatte für ein Cluster vorstrukturiert

Idee: Clusterwide System Disk

jedes Cluster Mitglied bootet in eine eigene Wurzel (root)

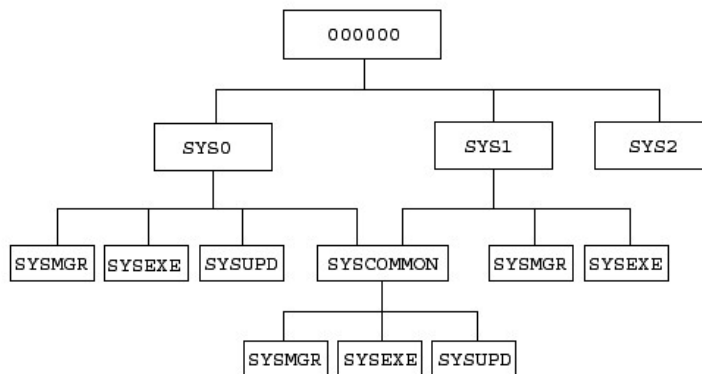
z. B.: SYS0, SYS1, SYS2, SYS10, SYS11, .....

Karl Bruns

9

## OpenVMS Logische Namen und Verzeichnisstrukturen

- Aufbau einer OpenVMS Systemplatte



Karl Bruns

10

## OpenVMS Logische Namen und Verzeichnisstrukturen

- Aufbau von SYSSYSROOT

OpenVMS wird verteilt in verschiedene Unterverzeichnisse unterhalb von SYSn:

SYSMGR, SYSEXE, SYSUPD, SYSHLP, SYSMMAINT, SYSLIB, ....

Nur: OpenVMS ist zu 90 % für alle Cluster Mitglieder gleich

Bsp.: SHOW.EXE, AUTOGEN.COM,  
Gegenbsp.: MODPARAMS.DAT, OPERATOR.LOG

Karl Bruns

11

## OpenVMS Logische Namen und Verzeichnisstrukturen

- Aufbau von SYSSYSROOT

Lösung: Verzeichnisebene SYSSSPECIFIC für einen bestimmten Knoten,  
SYSSCOMMON für sämtliche Knoten

Implementierung: Logische Namen !!!

Idee: Rekursivübersetzung

```
$ DEFINE A B          $ DEFINE A B,C,D,TEST.TXT
$ DEFINE B A          $ TYPE A
$ SHOW LOGICAL A
```

Karl Bruns

12

## OpenVMS Logische Namen und Verzeichnisstrukturen

- Definition der logischen Namen

beginne mit der Rekursivdefinition an oberster Stelle:

```
SYSSYSROOT = DKA0:[SYS0.],  
             DKA0:[SYS0.SYSCOMMON.]
```

damit:

```
SYSMANAGER = SYSSYSROOT:[SYSMGR]  
SYSSYSTEM  = SYSSYSROOT:[SYSEXE]  
SYSUPDATE  = SYSSYSROOT:[SYSUPD]
```

Karl Bruns

13

## OpenVMS Logische Namen und Verzeichnisstrukturen

- Concealed Device
  - allerdings ist SYSSYSROOT ein logischer Name, der nicht auf ein Verzeichnis zeigt !!
  - damit die Zusammensetzung bei der Übersetzung funktioniert, muß dieser Name mit dem Attribut „concealed“ versehen werden
  - die Suchreihenfolge legt fest, dass zuerst im spezifischen Bereich gesucht wird und falls die Datei dort nicht gefunden wird erst dann im allgemeinen Bereich gesucht wird

```
Bsp.: SYSSYSTEM:SYSUAF.DAT  
      SYSMANAGER:SYSTARTUP_VMS.COM
```

Karl Bruns

14

## OpenVMS Logische Namen und Verzeichnisstrukturen

- Concealed Device für die Benutzer

Ordne die Benutzerverzeichnisse nach Gruppen und lege ein concealed device darüber

```
$ CREATE/DIR DKA100:[PROJEKT1]
```

```
$ DEFINE/SYSTEM/EXEC/TRANS=CONCEALED  
DISK_PROJEKT1 DKA100:[PROJEKT1.]
```

```
$ CREATE/DIR DISK_PROJEKT1:[HUGO]/OWNER=HUGO
```

Karl Bruns

15

## OpenVMS Logische Namen und Verzeichnisstrukturen

- Concealed Device für die Benutzer

```
$ RUN AUTHORIZE  
UAF> add HUGO /device=DISK_PROJEKT1  
/ directory=HUGO/.....
```

Damit sieht Benutzer als Standardverzeichnis:

```
DISK_PROJEKT1:[HUGO]
```

Karl Bruns

16